Non-Parametric Multi-Scale Curve Smoothing

Paul Rosin

Cognitive Systems Group
School of Computing
Curtin University of Technology
Perth, 6001
Western Australia
email: rosin@cs.curtin.edu.au

ABSTRACT

Lowe[7] demonstrated a method for automatically segmenting and smoothing image curves by varying degrees. It was intended to remove noise and unnecessary fine detail, aiding subsequent processing such as grouping and matching. An alternative technique is described in this paper that is based on recursively subdividing the curve into alternative sets of sections. Rather than use thresholds on the values of curvature and its derivatives to determine the segmentation and degree of smoothing our technique is driven by three qualitative measures: 1) a criterion for selecting potential breakpoints, 2) a criterion for determining the amount of smoothing for curve sections, 3) a significance measure that determines which sections form the best selection. The advantages of the technique are robustness, scale invariance, and the absence of parameters.

1. INTRODUCTION

Many techniques and applications within computer vision involve the processing of image curves, which are usually derived through edge detection. The curves can either be matched directly with model boundaries or are first approximated by features such as lines, arcs, corners, etc. A problem with processing raw image curves is that they often contain noise and other undesirable fine detail. This can arise from two sources. First, noise, quantisation, and other errors occur during image acquisition and during the various stages of the extraction of the image curves from the image. Second, object boundaries are affected by neighbouring objects and the surrounding background. Different textures and grey level contrasts will alter the certainty and accuracy of boundary detection. Unwanted structures can be eliminated by smoothing, but this involves a trade-off between removing noise and removing useful detail, leading to the question of how much smoothing should be performed. Ideally, the correct amount of smoothing should remove as much noise as possible while retaining as much real structural detail as possible. Since errors can come from several sources they may not be uniform across different curves or within a curve, and may have differing amplitudes and types. Thus different curves and different sections of a curve may require different amounts of smoothing.

A disadvantage of many curve representations is that they are extremely sensitive to any noise. Smoothing is therefore a useful preprocessing stage. Some well-known examples of representations are the Medial Axis Transform (MAT)[3], codons[5], and Fourier Descriptors[2]. With the MAT even a tiny perturbation in the curve can radically change the structure of the medial axis, producing large unwanted

0-8194-1200-7/93/$4.00

branches. Codons are defined by the singularities of curvature. Without smoothing either the co-ordinates of the curve or the curvature function itself many spurious singular points of curvature arise, resulting in the curve being over-segmented into meaningless fragments. Finally, a common parameterisation of a curve for analysis by the Fourier Transform is by curvature and pathlength. The effects of errors arising just from quantisation can be significant, and are analysed by Bennet and MacDonald[2].

When deciding at what scale to perform curve smoothing there are three main approaches. The most straight forward is to represent the curves over a fixed range of scales[1,4,8,9]. Although this is a robust approach it produces a vast amount of data. A second approach is to represent each curve at only a small number of interesting or significant scales which we have called their natural scales[10]. Between each natural scale there is some qualitative change, e.g. they contain different structures. In contrast, the traditional multi-scale methods contain much redundancy since there will be little qualitative change between most curves at adjacent scales. This duplication of information makes the representation inefficient and cumbersome to process since the most relevant information is not made explicit. Finally, the curve can be represented by a single smoothed version. Either the whole curve is smoothed uniformly at a single scale or by different amounts over individual sections. The latter approach is more appropriate since curves often contain differently sized structures. There is a trade-off between completeness and conciseness that is balanced differently by the three approaches. The multi-scale is the most robust, guaranteeing curves will be represented at the correct scale (assuming adequate sampling of the scale dimension) but produces a proliferation of data. Natural scales significantly reduce the amount of data at the cost of occasionally missed or redundant scales. A single scale is the most compact, but necessarily can lose important information if a curve contains several superimposed structures at different scales.

Lowe[7] described a method for automatically segmenting and smoothing image curves that produces a single curve smoothed by different amounts over different sections. This report describes an alternative approach to perform the same function. Its advantages are robustness, scale invariance, and the absence of parameters. We first briefly describe Lowe's algorithm. The new algorithm is then introduced, and finally some comparative examples are provided.

2. LOWE'S ALGORITHM

Lowe's algorithm segments curves based on the change in curvature. A curve is smoothed through a range of octave separated scales with σ going from 1 to 8. At each scale the curve is split into sections in which all the points have a change in curvature below a threshold normalised by σ. A minimum length threshold of 2σ is set to prevent zero crossings of κ' near corners from being considered as short smooth intervals. The sections at all the scales are selected so that each point is represented by a single section. The following procedure is carried out to maximise the lengths of individual sections. All sections are considered and the longest section is chosen. The portions of all other sections that overlap the selected sections are removed, and the process is repeated until as much of the curve is covered as possible.

Although the algorithm works fairly well, there are several points to note. First, the reason given for segmenting the curve based on the change in curvature was that these represent likely points at which two objects cross. Large curvature difference is used rather than large curvature to partially overcome the effects of noise and scale. However, we have earlier noted that object boundaries may be locally affected by noise and the effects of different textures and contrasts of surrounding backgrounds. This implies that a variable amount of smoothing (and therefore segmentation) may be required *within* a section arising from a single object. These segmentation points will not be detected by analysing change in curvature.

A second consideration is the requirement for thresholds. Different values of the threshold on the change in curvature will produce different results, requiring user "tuning". The threshold on the change in curvature is normalised by σ so that the same threshold can be applied to the curve at the different levels of smoothing. However, it is still constant for different curves. This means that the technique cannot be scale independent since if a curve decreases in size its curvature proportionally increases.

3. OUR PROPOSED ALGORITHM

To overcome these limitations of Lowe's algorithm we have developed an alternative technique for segmenting and smoothing curves. It is based on recursively subdividing the curve into smaller sections which are smoothed by suitable amounts. We call the set of sections the *segmentation tree*. It contains all the possible alternative segmentations and representations of the image curve given the chosen breakpoints. From this tree the set of sections that make up the best unique description of the curve is chosen. The same general approach to segmentation and approximation by some feature(s) was taken by Lowe[6] for finding polygonal approximations of curves, and by Rosin and West[13,14] for segmenting curves into straight lines, circular arcs, and elliptical arcs.

3.1 Scale Selection

Each section of curve (to be defined by the breakpoints) is smoothed. Rather than smoothing by an arbitrary amount some qualitative criterion is required. Initially, the highest ranking (i.e. the most significant) natural scale was considered. It could be detected using the approach described by Rosin[10]. However, this idea was abandoned for several reasons. First, it would be possible for the most significant natural scale of a section of curve to be smaller than that of some of its subdivided sections. This would not agree with the general spirit of the segmentation algorithm which assumes that fits to large sections are coarser than fits to smaller sections. Second, determining and quantifying the natural scales is not totally robust, and sometimes several spurious natural scales are detected, confusing the rankings. Lastly, some curves do not contain a predominant natural scale. Fractal curves such as the Sierpinski triangle and the Koch snowflake are examples composed of repetitive structures, equally spaced over a range of scales, that would all be detected as equally significant.

Instead of employing natural scales a simpler and more appropriate method is used to determine the degree of smoothing. Each section of curve is smoothed by the amount necessary to eliminate all zero-crossings of curvature. This results in a simple section with no inflection points, and is either entirely concave, convex, or straight. This closely follows the earlier versions of the segmentation algorithm that fit features (e.g. analytic functions such as straight lines, circular arcs, and elliptical arcs) to curve sections. In this case the feature is a concave/convex/straight arc. Since the amount of smoothing required is not known in advance the curve section is repeatedly smoothed with increasing values of σ until no zero-crossings of curvature remain. Thus, for a given curve, the longer the sections considered in the segmentation tree the greater the degree of smoothing is likely to be required. When the segmentation tree is formed the higher levels of the tree will contain coarser features and the lower levels will contain finer features, producing in effect a multi-scale representation similar to Ballard's strip-tree[1].

3.2 Segmentation Tree Traversal

After the segmentation tree is generated the best set of features is selected from it. Each section is assigned a significance value. This is a quasi-psychological measure that describes the goodness of the representation. It is defined as the maximum deviation between the approximation and the original data normalised by the length of the feature. Since the original data and the smoothed version have a pointwise correspondence this is simply the maximum pointwise distance divided by the length of the curve. Normalising by the length has two functions. It can be seen as allowing longer (and therefore more perceptually significant) features to contain greater deviations from the norm. More importantly, it also ensures that the technique is scale independent. Favouring long sections was also employed in a more explicit manner by Lowe[7]. The best set of features is selected from the segmentation tree by traversing upwards from the leaves to the root. At each node the feature is retained if it is more significant than all its children, otherwise it is replaced by its children. After traversing the tree the remaining feature curves constitute the best representation of the curve.

3.3 Breakpoint Determination

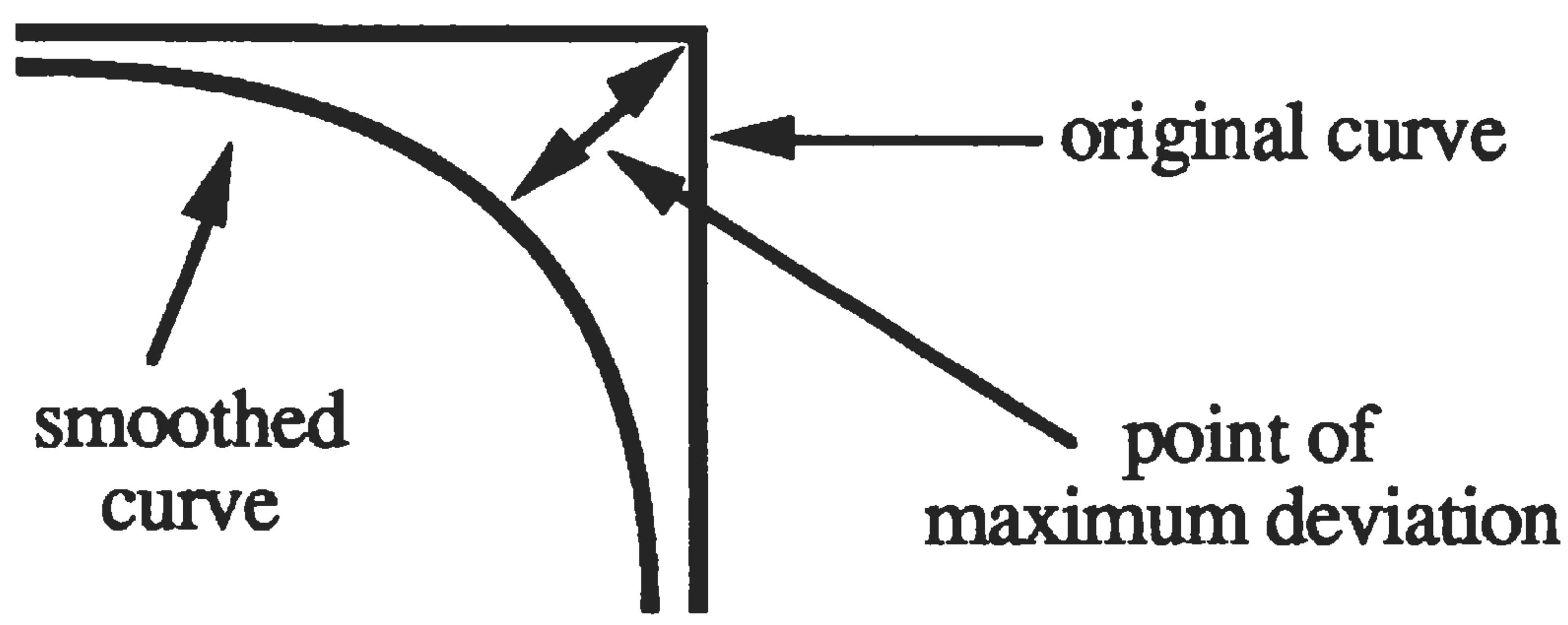

figure 1 - point of maximum deviation as a breakpoint

There are a variety of approaches to choosing the breakpoint to split each curve segment into two. We describe here three techniques, all of which are non-parametric. The first follows the original segmentation and approximation algorithms which used the point of maximum deviation between the original data and the approximation as the breakpoint. In many cases this will correspond to the point of maximum change in curvature. An example is shown in figure 1.

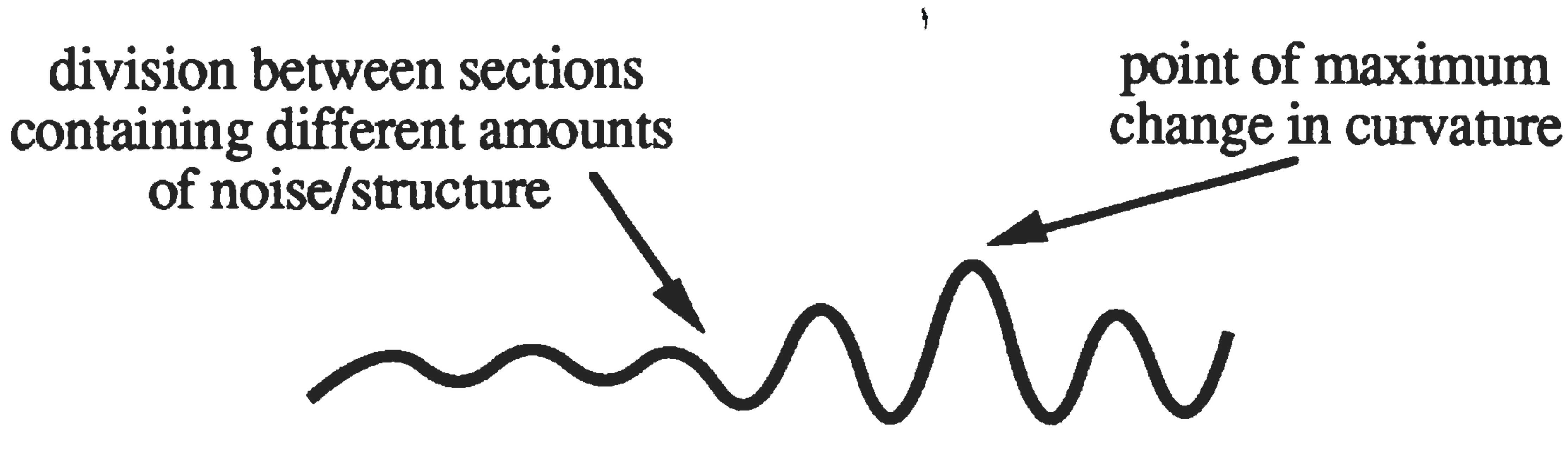

figure 2 - change in texture as a breakpoint

This approach has limitations in common with Lowe since large deviations, or large changes in curvature will not identify changes in the texture of a curve. By texture we mean the characteristics of the underlying noise and/or fine detail. An example of a curve containing two levels of detail is shown in figure 2. The ideal segmentation point that separates the two sets of structures does not correspond to the point of maximum deviation (or maximum change in curvature).

The second technique addresses the curve texture by analysing the heterogeneity of the two sections of the curve on either side of the potential breakpoint. In other words, the sections should have different noise characteristics. The best breakpoint will be the point at which the noise characteristics of the two sections differ most. Heterogeneity is calculated by comparing the average errors between the original data and its smoothed version for the two sections. At every point along the curve the average error on the section to the left ($error_{left}$) and to the right ($error_{right}$) of the point is calculated. Heterogeneity is then calculated as:

$$heterogeneity = \begin{cases} 1 - \dfrac{error_{left}}{error_{right}} & \text{if } error_{left} \leq error_{right} \\ 1 - \dfrac{error_{right}}{error_{left}} & \text{if } error_{left} > error_{right} \end{cases}$$

However, this measure could result in breakpoints being chosen very close to the ends of the curve. To avoid this another factor is included to bias the breakpoint towards the centre of the curve, where $length_{left}$ and $length_{right}$ are the lengths of the curve sections to the left and right of the potential breakpoint:

$$centrality = \begin{cases} \dfrac{length_{left}}{length_{right}} & \text{if } length_{left} \leq length_{right} \\ \dfrac{length_{right}}{length_{left}} & \text{if } length_{left} > length_{right} \end{cases}$$

Both these measures produce a response between 0 and 1, corresponding to worst and best results. They are combined into a single measure, also ranging from 0 to 1, as:

$$measure = heterogeneity \times \sqrt{centrality}$$

The square root of the centrality measure is used since it was otherwise found to dominate the combined measure.

Although this measure uses the correct principles to determine the breakpoint, it unlikely to be completely robust. Real data does not display such well defined noise characteristics. In addition, when a curve contains more than two levels of structure the breakpoint determination will not work well unless two levels dominate. Therefore, a third approach is considered, which is to subdivide the curve at its midpoint. While it does not take the curve's structure into account it does have the benefits of robustness and efficiency.

3.4 Curve Smoothing

The curve is represented as two coordinate functions of a path length parameter t:

$x = x(t)$ and $y = y(t)$

To smooth the curve these functions are independently convolved with a Gaussian function of standard deviation σ:

$$G_\sigma(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2}$$

Curvature is calculated using the convolution of the first and second derivatives of the Gaussian:

$$\kappa = \frac{X'Y'' - Y'X''}{\left(X'^2 + Y'^2\right)^{3/2}}$$

The Gaussian has the property that it does not introduce zero-crossings with increased smoothing, which is required for the selection of the degree of smoothing for curve segments to be well conditioned. However, there are several problems associated with curve smoothing which have to be addressed. The first is the tendency of the smoothed curve to shrink towards its centre of curvature. As this effect is proportional to the degree of smoothing as well as the degree of curvature it can become pronounced for the large amounts of smoothing that may be required to eliminate kinks. Fortunately, a solution has been provided by Lowe[7] to correct the shrinkage, which is employed here.

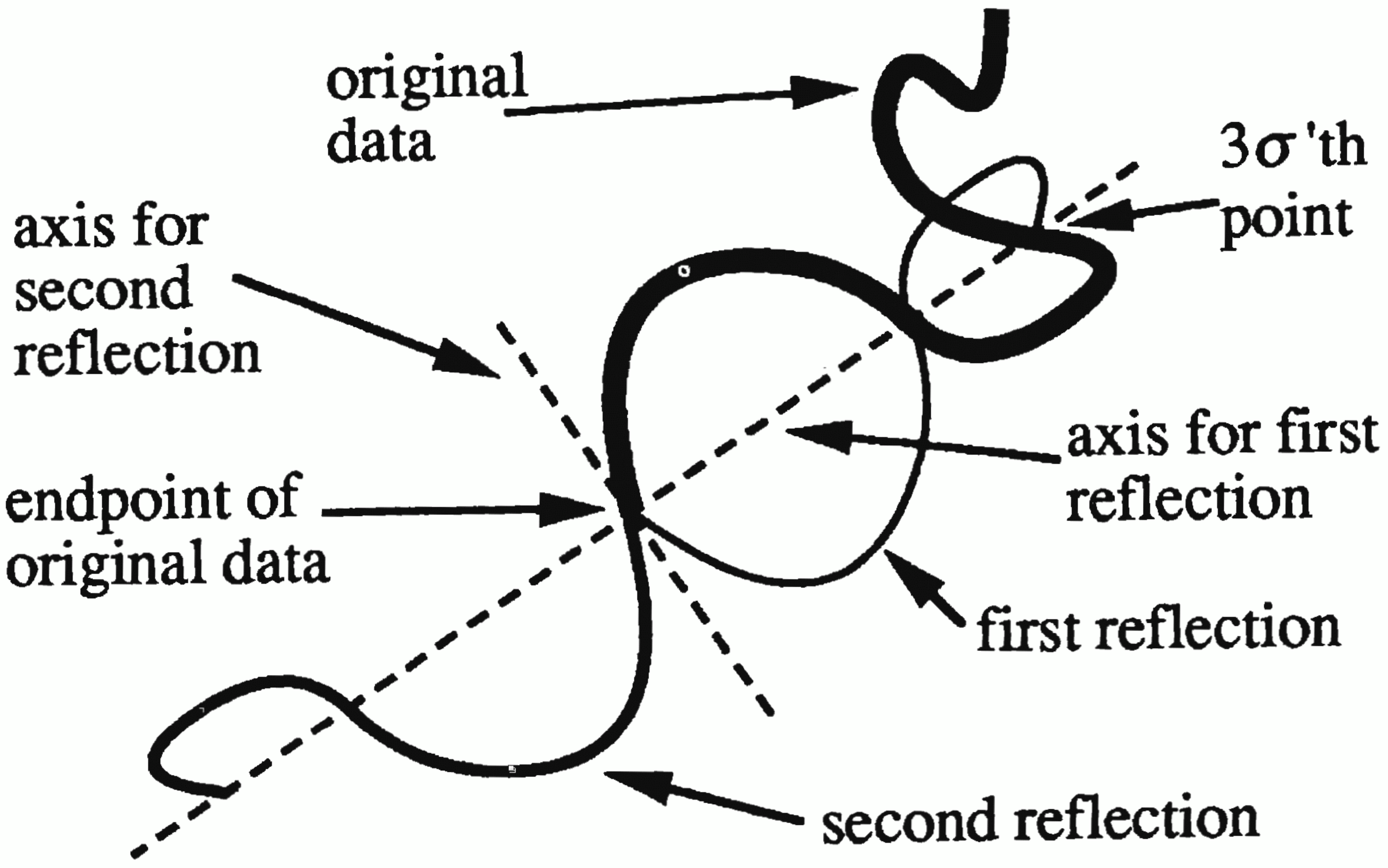

figure 3 - extending a curve beyond its endpoints

The second problem concerns the need to convolve the Gaussian mask with the data when it extends beyond the end of the curve. The convolution kernels are defined over an infinite range but can be safely truncated at a distance of 3σ from their centre. For closed curves $x(t)$ and $y(t)$ are periodic functions and the data wraps around. If the curve is open then a way must be found to extend the data beyond the end of the curve. There are several solutions possible; we have chosen to extend the data by 3σ points at each end of the curve by duplicating and reflecting the curve. This is most often done by reflecting the curve about a line passing through the endpoint that is normal to the tangent estimated at the closest curve point for which a reliable estimate is available[7]. We avoid the difficulty in estimating the tangent by first reflecting about the straight line between the endpoint to the previous 3σ'th point and then reflecting again about the normal to that line at the curve endpoint (see figure 3). In practise this is performed by a single transformation. The advantages of this technique are that it is robust and non-parametric. Points can be reliably reflected even for very noisy curves with a high rate of change of curvature.

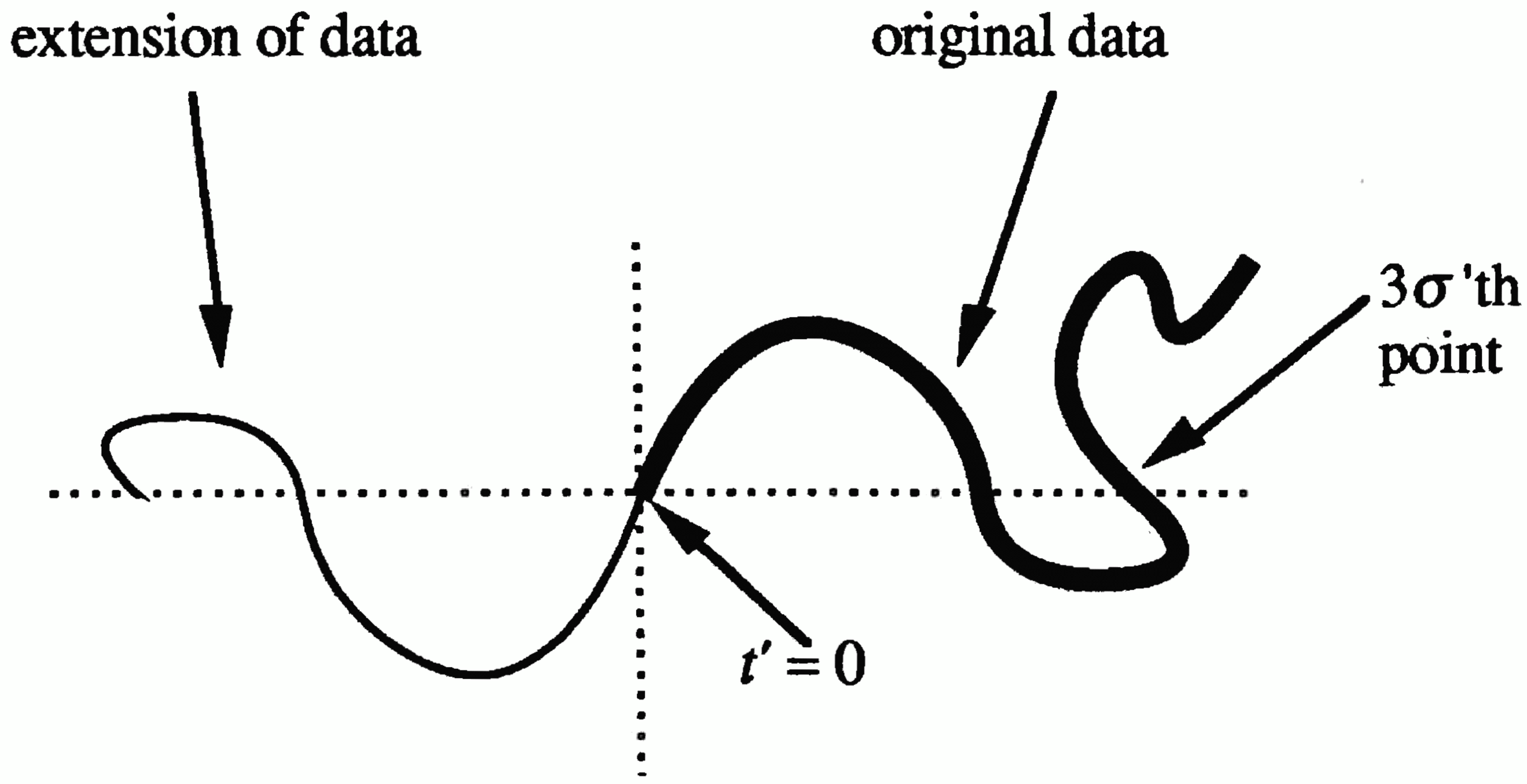

figure 4 - the curve extension ensures smoothed endpoints do not drift

A third desirable property is that the location of the curve endpoint is preserved after smoothing. This can be demonstrated without loss of generality by setting (or rotating) the extended curve so that the lines of reflection lie on the co-ordinate axes (figure 4). Setting the pathlength parameter of the curve t' to zero at the end of the original data (which lies on the origin) it can be seen that all the extended data points $(x_{-t'}, y_{-t'})$ have symmetric pairs $(x_{t'}, y_{t'})$, with $x_{-t'} = -x_{t'}$ and $y_{-t'} = -y_{t'}$. Thus, the averaged co-ordinate values at the endpoint remain (0, 0). Towards the end of the curve the smoothed points approach the horizontal axis and the smoothed curve becomes flattened. Retaining the curve endpoints after smoothing is important since it ensures that even if sections of a single curve are smoothed by different amounts the result will still be continuous (although only piecewise smooth). In contrast, Lowe's method allows the endpoints to drift significantly. This is apparent in the examples which show that curves are segmented into discontinuous segments.

4. RESULTS

Figure 5 shows the edges extracted from an image using the Canny edge detector. The results of smoothing these edges using Lowe's algorithm are shown in figure 6, where the thickness of the lines is proportional to the amount of smoothings. Figures 7a-c show the results of our algorithm using the following breakpoint criteria: a) maximum deviation, b) homogeneity, and c) midpoint. A considerable difference can be seen between these approaches. None the less, it can be seen that even without manually set thresholds they provide reasonable results. In particular the technique for extending curves has produced continuous curves in comparison to the occasionally fragmented output from Lowe's algorithm.

The outline of Africa is used to demonstrate the properties of scale invariance. It was subsampled to provide three versions at different sizes as shown in figure 8a. The results of Lowe's algorithm are given in figure 8b, and our algorithm using the midpoint breakpoint criterion in figure 8c. The latter results can be seen to be more pleasing and consistent. Since the curves are only analysed at pixel resolution, creating quantisation errors for the two smaller curves, the results are not exactly scale-invariant. Of course, similar sampling problems would occur with real data.

5. CONCLUSIONS AND FUTURE WORK

We have described a method for segmenting and smoothing curves by varying degrees. It is based on an algorithm previously used for segmenting curves into a variety of other features such as straight lines and arcs. Its advantages are that it is relatively robust, scale-invariant, and non-parametric. This means that it can be applied to a variety of curves with different noise, scale and structural characteristics and not require any user tuning to ensure good performance.

The algorithm requires sections of curves to be smoothed by various amounts. Gaussian smoothing with shrinkage correction is applied until all zero-crossings of curvature are eliminated. In the future it would be interesting to experiment with other smoothing methods and alternative criteria for determining the suitable amounts of smoothing. For instance, Gaussian smoothing causes sharp corners to become rounded, which is not always desirable. Adaptive smoothing techniques[16] could be employed which would retain such discontinuities. Other criteria for choosing the optimal scale could be based on cross-validation techniques[17] or bending energy[9].

The general segmentation algorithm has already been successfully applied to segment three-dimensional curves into lines and circular arcs[15]. Likewise, the current smoothing algorithm could be easily extended from 2D to 3D curves. Work is also being undertaken to apply the curve smoothing as a pre-processing stage for curve grouping[11] and representation by codons[12].

6. ACKNOWLEDGMENTS

I would like to thank David Lowe for providing his code for curve smoothing.

7. REFERENCES

1. D.H. Ballard, "Strip Trees: A Hierarchical Representation for Curves", *Comm. ACM,* Vol. 24, pp. 310-321, 1981.
2. J.R. Bennett, J.S. MacDonald, "On the Measurement of Curvature in a Quantized Environment", *Trans. IEEE Comp.*, Vol. 24, pp. 803-820, 1975.
3. H. Blum, R.N. Nagel, "Shape Description using Weighted Symmetric Axis Features", *Pattern Recognition*, Vol. 19, pp. 167-180, 1978.
4. G. Dudek, J.K. Tsotsos, "Shape Representation and Recognition from Curvature", *Proc. CVPR*, pp. 35-41, 1991.
5. D.D. Hoffman, W.A. Richards, "Representing Smooth Plane Curves for Recognition", *Proc. AAAI*, pp. 5-8, 1982.
6. D.G. Lowe, "Three-Dimensional Object Recognition from Single Two-Dimensional Images", *AI,* Vol. 31, pp. 355-395, 1987.
7. D.G. Lowe, "Organization of Smooth Image Curves at Multiple Scales", *Proc 2nd ICCV*, pp. 558-567, 1988.
8. D.H. Marimont, "A Representation for Image Curves", *Proc. AAAI*, pp. 237-242, 1984.
9. P. Meer, E.S. Baugher, A. Rosenfeld, "Extraction of Trend Lines and Extrema from Multiscale Curves", *Pattern Recognition,* Vol. 21, pp. 217-226, 1988.

10. P.L. Rosin, "Representing Curves at their Natural Scales", Pattern Recognition, 1992.

11. P.L. Rosin, "Grouping Curved Lines", Technical Report No. 1, School of Computing Science, Curtin University of Technology, 1992.

12. P.L. Rosin, "Contour Matching using Codons", *Proc. Int. Conf. on Image Processing*, Singapore, pp. 221-225, 1992.

13. P.L. Rosin, G.A.W. West, "Segmentation of Edges into Lines and Arcs", *Image and Vision Computing*, Vol. 7, pp. 109-114, 1989.

14. P.L. Rosin, G.A.W. West, "Segmenting Curves into Elliptic Arcs and Straight Lines", *Proc. 3rd ICCV*, pp. 75-78, 1990.

15. P.L. Rosin, G.A.W. West, "Non-Parametric Segmentation of 2D and 3D Curves into Lines and Arcs", *Proc. Int. Conf. on Automation, Robotics, and Vision*, Singapore, pp. 3.6.1-3.6.5, 1992.

16. P. Saint-Marc, J.-S. Chen, G. Medioni, "Adaptive smoothing: A general tool for early vision", *Trans. IEEE PAMI*, Vol. 13, pp. 514-529, 1991.

17. B. Shahraray, D.J. Anderson, "Optimal Estimation of Contour Properties by Cross-Validated Regularization", *Trans. IEEE PAMI*, Vol. 11, pp. 600-610, 1989.

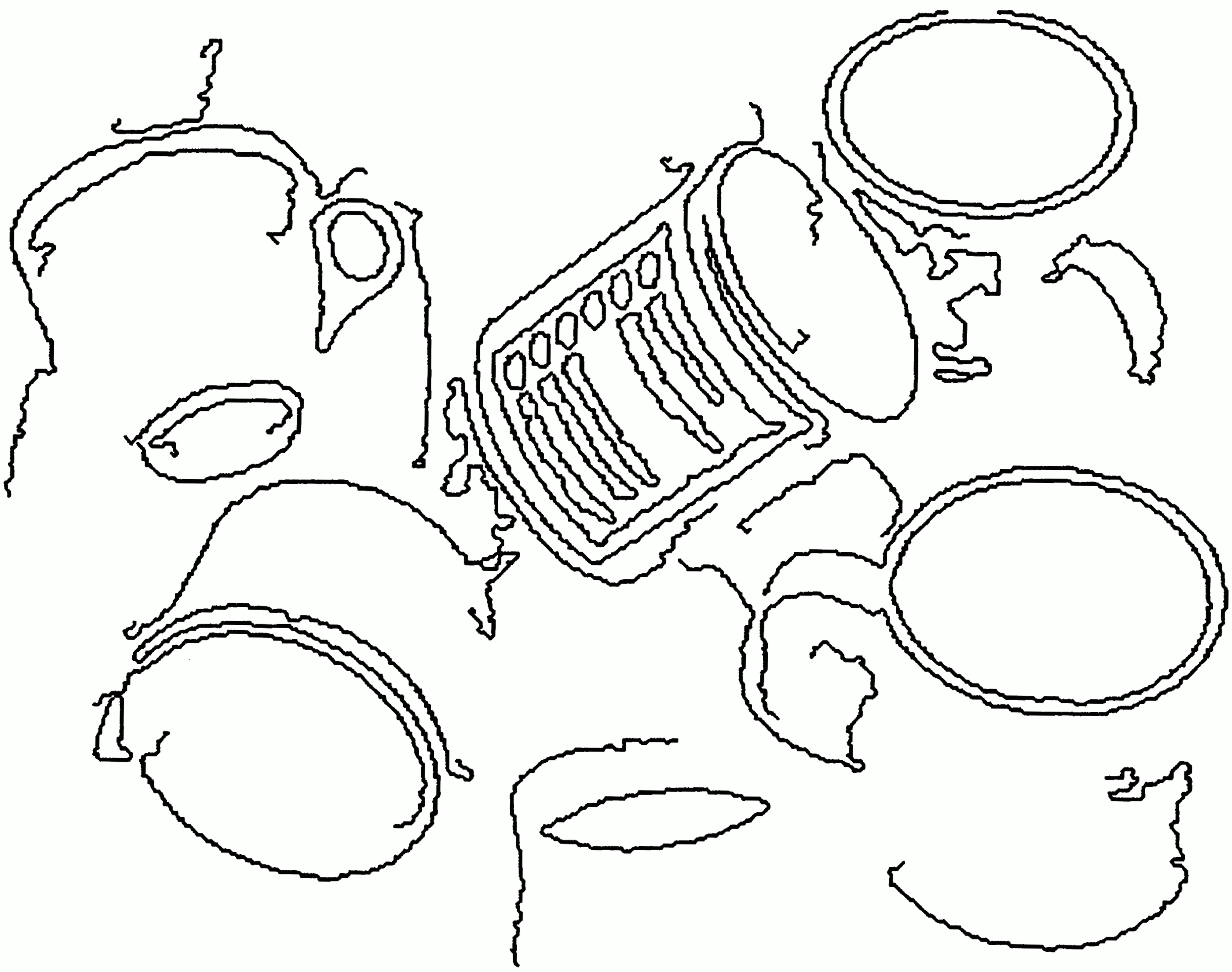

figure 5 - edges produced by Canny detector

figure 6 - smoothed curves produced by Lowe's algorithm

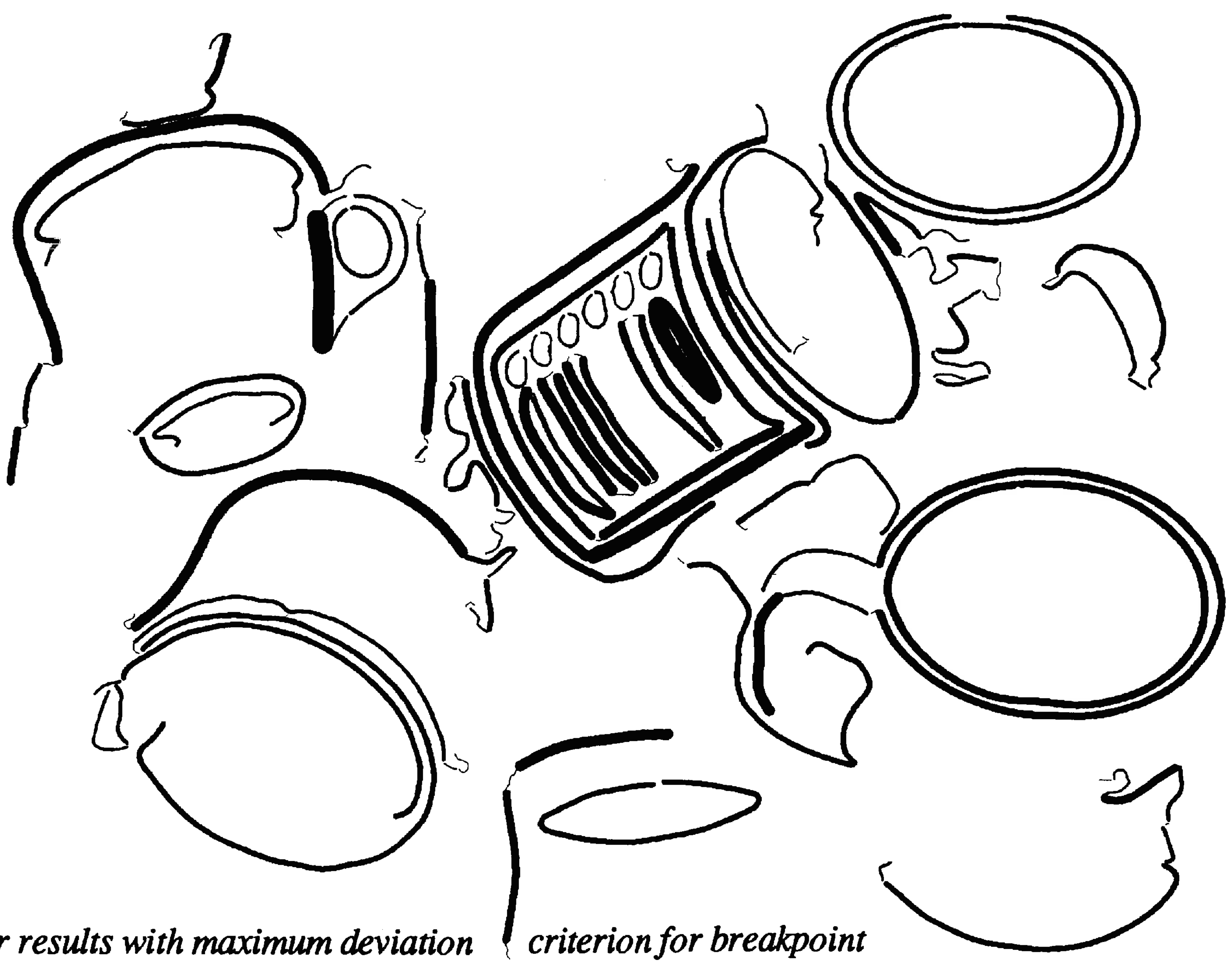

figure 7a - our results with maximum deviation criterion for breakpoint

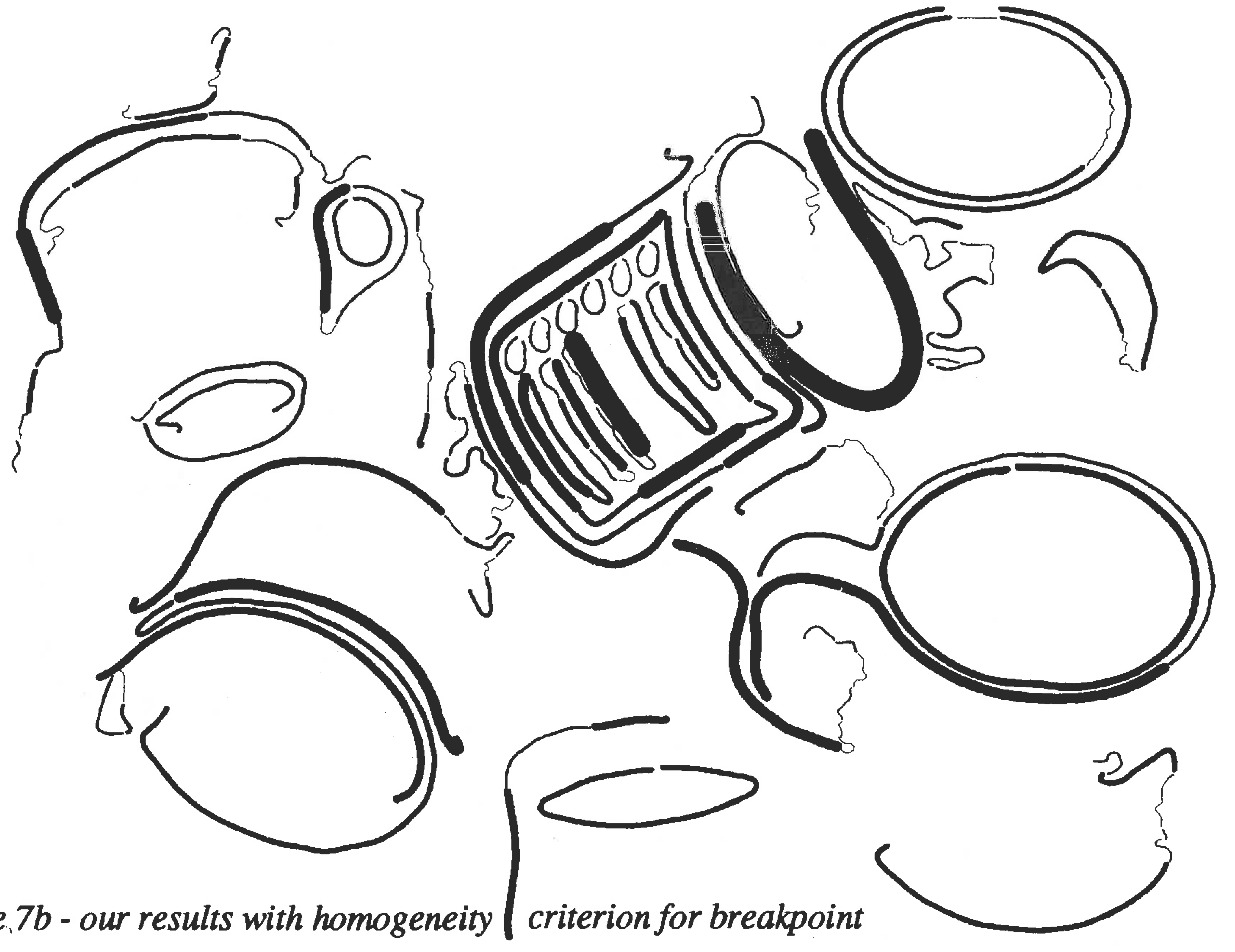

figure 7b - our results with homogeneity criterion for breakpoint

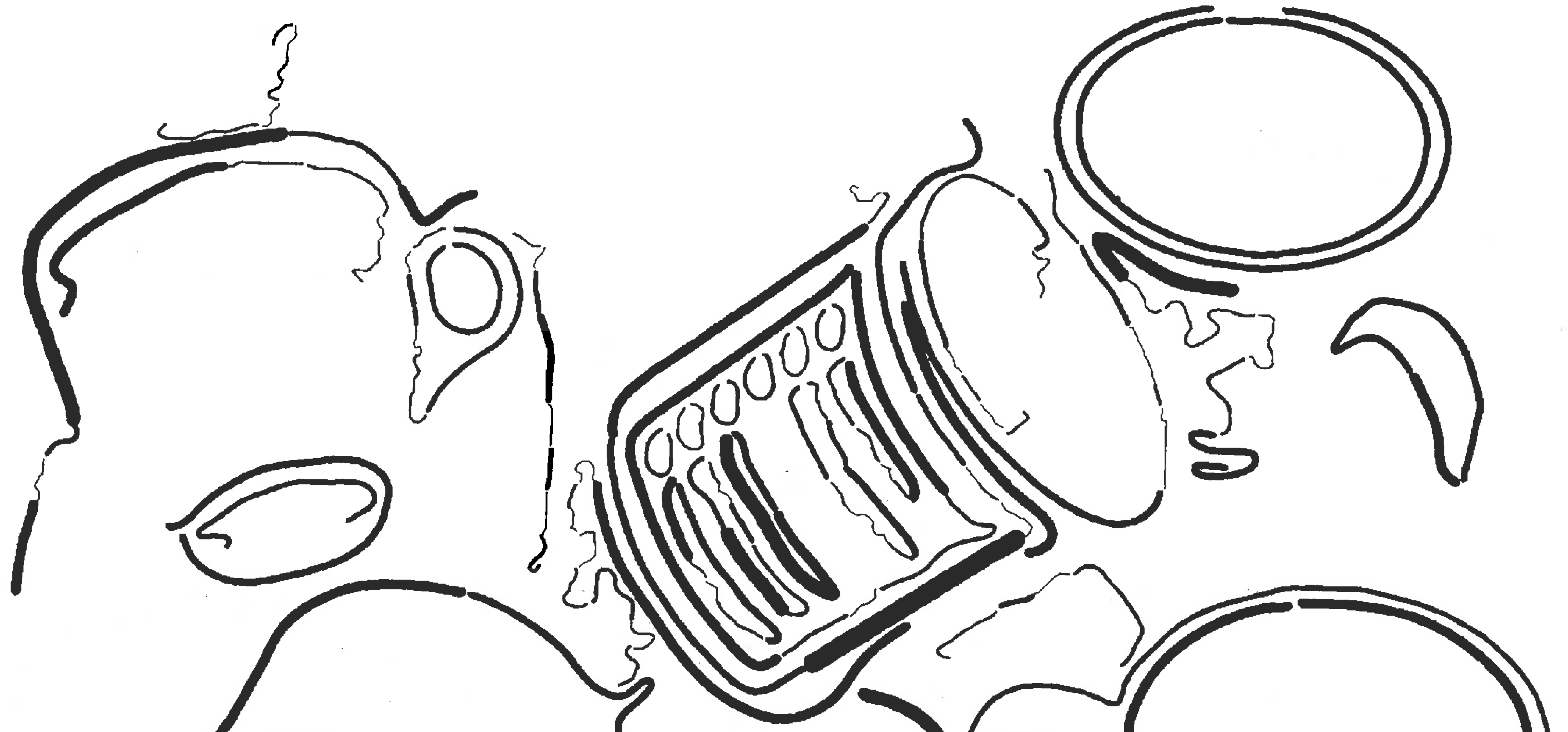

figure 7c - our results with midpoint criterion for breakpoint

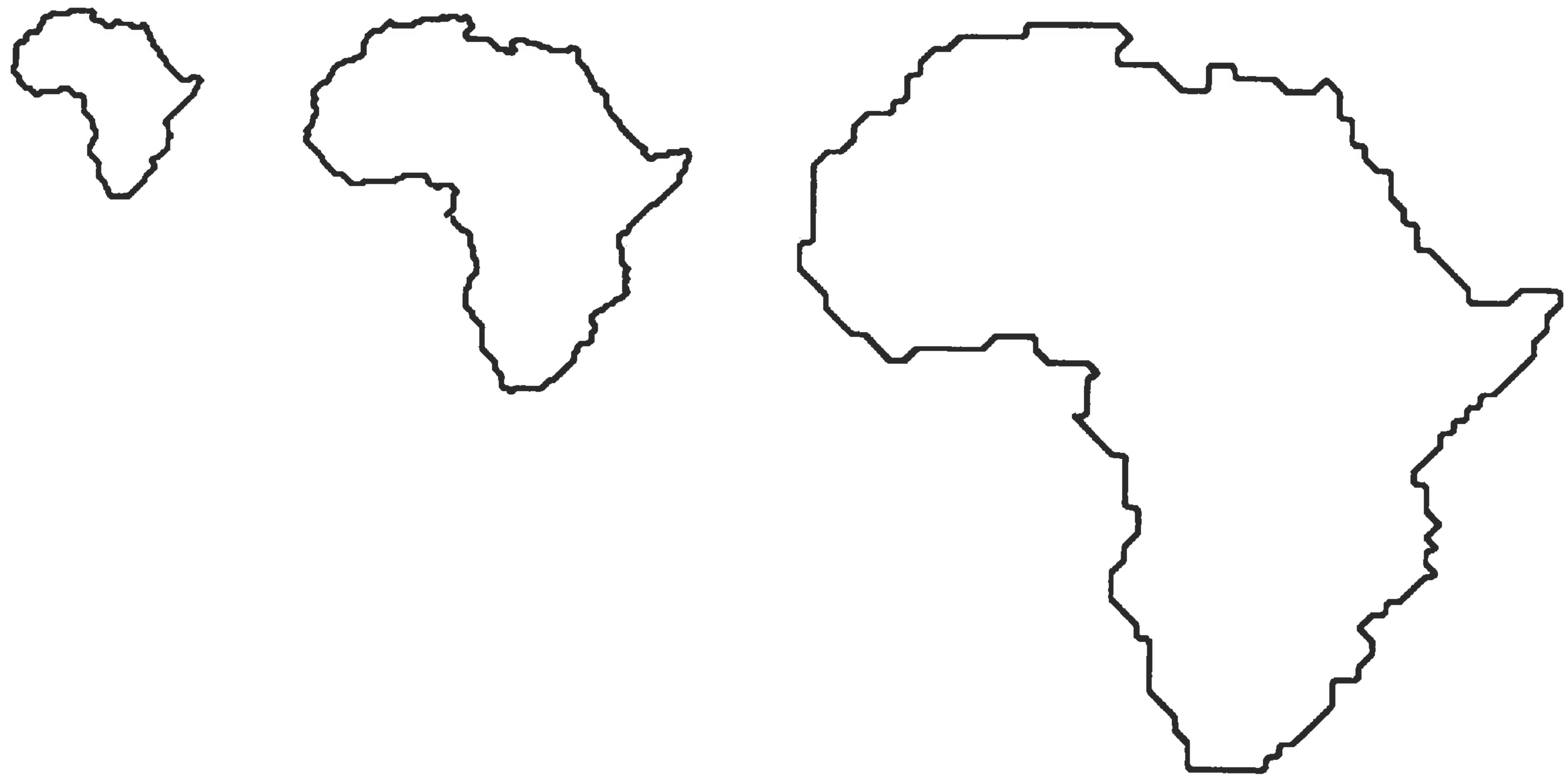

figure 8a - pixel data for outline of Africa at three scales

figure 8b - smoothed curves produced by Lowe's algorithm

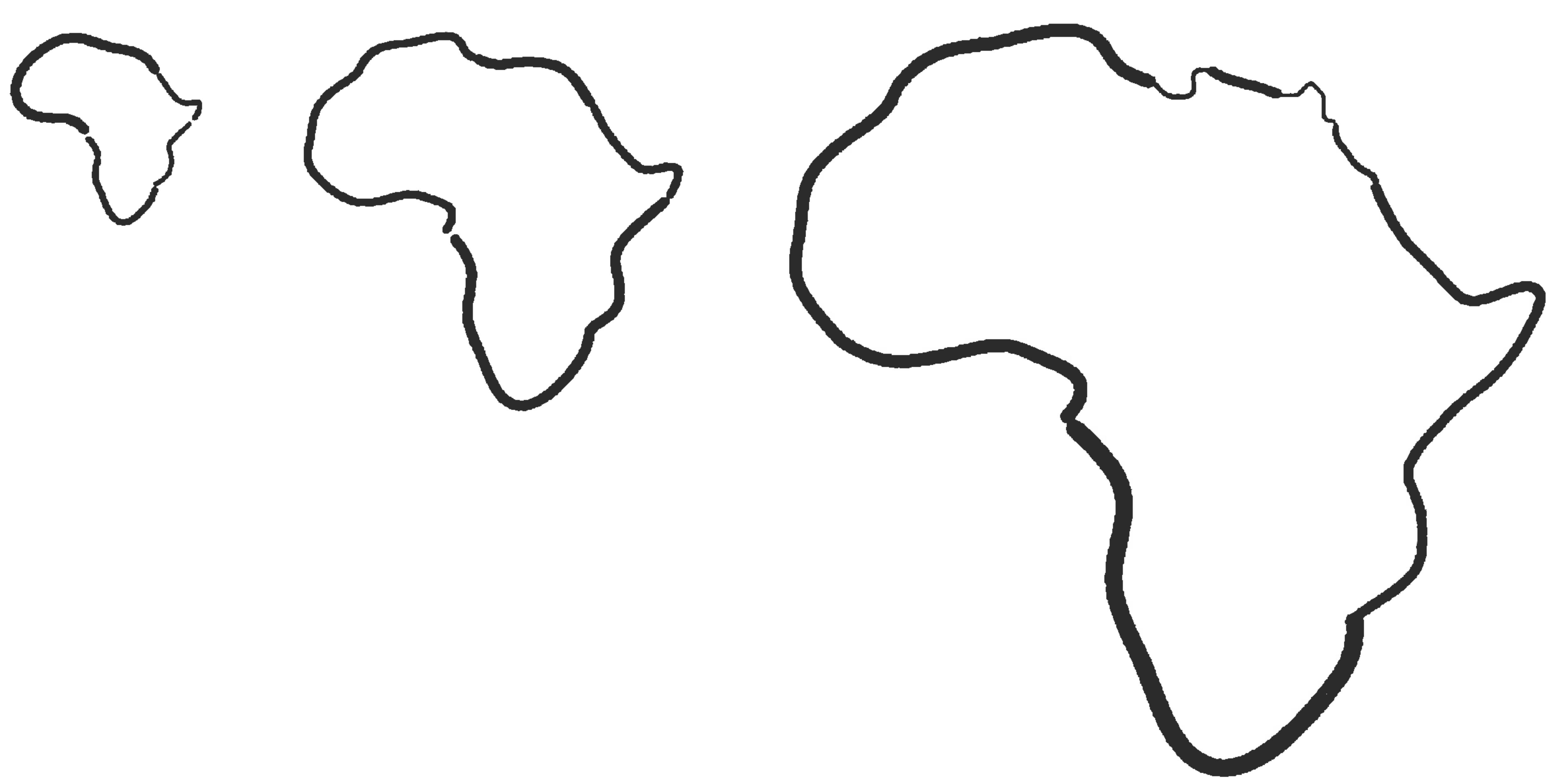

figure 8c - smoothed curves produced by our algorithm